IN THE CLAIMS:

Please amend claims 1-13 and add claims 14-20 as follows:

CLAIMS

1. (Currently Amended) A ~~M~~method of storing encrypted data in a ~~random access~~ memory, comprising the steps of:

~~in which data words, which each comprise a predetermined number of data bits, are storable, characterized in that, before storage, an~~ encrypti~~o~~ng ~~of each~~ data word by permutating each data bit of the data word using a permutation key to generate ~~(M) is effected whereby~~ a permutated data word; and

storing the permutated data word in the memory ~~(P) with a predetermined number of data bits is generated from each data word (M), or from a data word (M) derived from this data word, by one-to-one permutation of the individual data bits (M[n-1]-M[0]) using a first permutation key (Mp)~~.

2. (Currently Amended) The ~~M~~method of ~~according to C~~claim 1, ~~in which~~ where after the step of permutating, further comprising the step of substituting each ~~the individual~~ data bits ~~(M[n-1]-M[0])~~ of the permutated data word ~~(Mp) are substituted before storage~~ using a ~~first~~ substitution key to generate a substitute data word, and where the step of storing comprises the step of storing the substitute data word in the memory ~~in order to provide an encrypted data word (M')~~.

3. (Currently Amended) The ~~M~~method of ~~according to C~~claim 1, where the step of encrypting further includes the step of substituting each data bit of the unencrypted data word using a substitution key prior to the step of permutating to generate a substitute data word, and

where the step of permutating comprises the step of permutating each data bit of the substitute data word using the permutation key to generate the permutated data word~~in which before rearrangement the individual data bits of the data word (M) are substituted using a first substitution key (S) in order to provide a substituted data word~~.

4. (Currently Amended) The ~~M~~method ~~according to one~~ of ~~the foregoing~~ claims 1, where ~~in which~~ the permutation key ~~(P) has a number~~ includes a plurality of ~~unique~~ subkeys ~~(P[n-1]-P[0])~~ corresponding to the number ~~n~~ of the data bits of the data word, and where each one of the subkeys includes a plurality of key bits, where the step of permutating each data bit in the data word using a permutation key further comprises the steps of:

assigning each one of the subkeys to a corresponding one of the data bits of the permutated data word; and

mapping each data bit of the unencrypted data word to a corresponding one of the data bits of the permutated data word using the corresponding assigned subkey~~which subkeys are assigned to one data bit each (Mp[n-1]-Mp[0]) of the permutated data word (Mp), and which each indicate the data bit (M[n-1]-M[0]) of the data word to be permutated (M), which data bit is to be mapped to this data bit (Mp[n-1]-Mp[0]), wherein each subkey (P[n-1]-P[0]) comprises a number of key bits (P[n-1,m-1]-P[n-1,0], P[k,m-1]-P[k,0], P[0,m-1]-P[0,0])~~.

5. (Currently Amended) The ~~M~~method of ~~according to C~~claim 4, where ~~in which~~ the step of mapping comprises: ~~of a data bit (M[n-1]-M[0]) of the data word to be permutated (M) to a data bit (Mp[k]) of the permutated data word is effected incrementally using a subkey (P[k]) by the following steps:~~

a) selecting a first group of the data bits of the data word ~~to be permutated (Mp)~~ as determined by a first one of the plurality of key bits ~~(P[k,0])~~ of the corresponding assigned subkey ~~(P[k])~~;

b) selecting a second group of the data bits of the data word from the first group of the data bits ~~obtained by the previous selection~~ as determined by a second one of the plurality of key bits ~~(P[k,1])~~ of the corresponding assigned subkey ~~(P[k])~~; and

c) repeating step b), each time using an additional one of the plurality of key bits of the corresponding assigned subkey ~~(P[k,2]...P[k,m-1])~~ until there exists ~~selected group comprises only~~ one remaining ~~more~~ data bit of the data word, where the one remaining data bit ~~which~~ corresponds to the data bit ~~(Mp[k])~~ of the data word mapped to ~~of~~ the corresponding data bit of the permutated data word ~~(Mp)~~.

6. (Currently Amended) The ~~M~~method of ~~according to C~~claim 5, where ~~in which~~ the number of data bits ~~contained~~ in the second ~~a~~ group of the data bits of the data word is reduced ~~from one step to the next~~ by a factor of ~~2~~two from the number of data bits in the first group of the data bits of the data word, and where the number of data bits in each group of the data bits of the data word in each iteration of step c is reduced by a factor of two.

7. (Currently Amended) The ~~M~~method ~~according to one~~ of ~~the foregoing~~ claims 2, where ~~in which~~ the ~~first~~ substitution key ~~(S) has a number~~ includes a plurality of key bits ~~(S[n-1]...S[0])~~ corresponding to the number of data bits of the permutated ~~of the~~ data word ~~to be substituted (Mp)~~, where the step of substituting each data bit of the permutated data word using a substitution key further comprises the step of mapping ~~wherein~~ each data bit of the permutated

data word ~~to be substituted (Mp) is mapped unchanged or inverted~~ to a data bit ~~(M'[n-1]...M'[0])~~ of the substituted data word ~~(M')~~ in one of an unchanged form and an inverted form as determined by the corresponding one of the~~se~~ key bits ~~(S[n-1]...S[0])~~.

8. (Currently Amended) The ~~M~~method ~~according to one~~ of ~~the foregoing~~ claims 3, ~~in which the permutation key (P) and the substitution key (S) are regenerated before a rewriting to the memory after a deletion~~ where the substitution key includes a plurality of key bits corresponding to the number of data bits of the data word, where the step of substituting each data bit of the data word using a substitution key further comprises the step of mapping each data bit of the data word to a data bit of the substituted data word in one of an unchanged form and an inverted form as determined by the corresponding one of the key bits.

9. (Currently Amended) The ~~M~~method ~~according to one~~ of ~~the foregoing~~ claims 1, further comprising the step of ~~which in order to~~ generating~~e~~ the~~a~~ permutation key by the ~~(P) comprises~~ the following steps:

a) randomly generating a sub-permutation-key and assigning the generated sub-permutation-key to a data bit ~~position~~ of the permutated data word;

b) checking whether the generated sub-permutation-key has already been assigned to a ~~generated for another~~ data bit ~~position~~ of the permutated data word, and retaining the generated sub-permutation-key as the assigned sub-permutation-key if the generated sub-permutation key ~~it~~ has not yet been assigned to a data bit of the permutated data word~~generated, and rejecting the generated sub-permutation-key if it has already been generated~~; and

c) implementing steps a) and b) until a sub-permutation-key is assigned~~generated~~ to ~~for~~ each data bit ~~position~~ of the permutated data word~~(Mp)~~.

10. (Currently Amended) The ~~M~~method of claim 1, further comprising the step of decrypting the stored permutated data word using a second permutation key matched to the permutation key used to generate the permutated data word~~according to one of the foregoing claims, in which a data word (M'), generated from a data word (M) using the first key after being read out from the memory, is permutated in order to generate the data word using a second permutation key (P') which is matched to the first permutation key (P)~~.

11. (Currently Amended) A ~~D~~device that ~~to~~ encrypts and~~/~~ decrypts a data word ~~(M)~~ ~~comprising~~ having a predetermined number of data bits ~~(M[n-1], M[k], M[0])~~, the ~~which~~ device having ~~has~~ a permutation unit ~~(14) with~~ comprising~~the following features~~:

–a plurality of data inputs that receive ~~to supply~~ the data bits ~~(M[n-1], M[k], M[0])~~ of the data word~~to be permutated (M)~~; and

~~–outputs to supply the data bits (Mp[n-1], Mp[k], Mp[0]) of a permutated data word (Mp) of the predetermined length (n);~~

~~–permutation key inputs to supply a permutation key (P) which comprises a number (n) of subkeys (P[n-1]... P[0]) corresponding to the number of data bits~~;

–a ~~number~~ plurality of selection units ~~(14_n-1, 14_k, 14_0)~~ corresponding to the number of data bits of the data word, where~~to which selection units~~ each one of the selection units is responsive to a ~~one~~ subkey portion of a permutation key~~each is assigned~~, where each one of the selection units ~~and which~~ provides one data bit each ~~(Mp[n-1], Mp[k], Mp[0])~~ of a~~the~~ permutated data word ~~(Mp)~~ from the corresponding data bit of the data word as determined by the corresponding one ~~each~~ of the subkeys ~~(P[n-1]... P[0]) from the data bits of the data word to be permutated (M)~~.

12. (Currently Amended) The ~~D~~device of ~~according to C~~claim 11, where ~~in which~~ each ~~of the~~ selection units ~~(14_k)~~ comprises ~~has~~ a number of consecutively arranged selection stages ~~(141_n-1, 141_k, 141_0)~~ corresponding to a~~the~~ number of permutation key bits of the corresponding subkey for that selection unit, where~~in~~ a first selection stage ~~(141_0) is designed,~~ is responsive to a first one of the permutation key bits ~~as determined by a first key bit (P[k,0]),~~ to select and provide a first group of data bits of~~from~~ the data word ~~to be permutated (M)~~, and where~~in~~ subsequent ones of the selection stages ~~(141_1, 141_2, 141_m-1) are designed, in each case as determined by~~ are each responsive to subsequent ones of the permutation key bits~~a key bit (P[k,1], P[k,2], P[k,m-1]),~~ to select a subgroup of the data bits from a~~the~~ group of data bits of the data word provided by the respective previous selection stage.

13. (Currently Amended) The ~~D~~device of ~~according to C~~claims 11 ~~or 13~~, further comprising a ~~in which~~ a substitution unit ~~is~~ connected ~~before or~~ after the permutation unit ~~(14)~~, that~~which substitution unit~~ substitutes each data bits ~~(Mp[n-1], Mp[k], Mp[0])~~ of the permutated ~~a~~ data word ~~to be substituted (Mp) as determined by~~ in response to a substitution key ~~(S)~~.

14. (New) The device of claim 11, further comprising a substitution unit connected before the permutation unit, that substitutes each data bit of the data word in response to a substitution key.

15. (New) A method of storing encrypted data in a memory, comprising the steps of:

encrypting a data word by permutating each data bit of the data word using a permutation key to generate a permutated data word;

substituting each data bit of the permutated data word using a substitution key to generate a substitute data word; and

storing the substitute data word in the memory.

16. (New) The method of claim 15, where the permutation key includes a plurality of subkeys corresponding to the number of the data bits of the unencrypted data word, and where each one of the subkeys includes a plurality of key bits, where the step of permutating each data bit further comprises the steps of:

assigning each one of the subkeys to a corresponding one of the data bits of the permutated data word; and

mapping each data bit of the data word to a corresponding one of the data bits of the permutated data word using the corresponding assigned subkey.

17. (New) The method of claim 16, where the step of mapping comprises the following steps:

a) selecting a first group of the data bits of the data word as determined by a first one of the plurality of key bits of the corresponding assigned subkey;

b) selecting a second group of the data bits of the data word from the first group of the data bits as determined by a second one of the plurality of key bits of the corresponding assigned subkey; and

c) repeating step b), each time using an additional one of the plurality of key bits of the corresponding assigned subkey until there exists one remaining data bit of the data word, where the one remaining data bit corresponds to the data bit of the data word mapped to the corresponding data bit of the permutated data word.

18. (New) A method of storing encrypted data in a memory, comprising the steps of:

substituting each data bit of an unencrypted data word using a substitution key to generate a substitute data word; and

permutating each data bit of the substitute data word using a permutation key to generate a permutated data word;

storing the permutated data word in the memory.

19. (New) The method of claim 18, where the permutation key includes a plurality of subkeys corresponding to the number of the data bits of the substitute data word, and where each one of the subkeys includes a plurality of key bits, where the step of permutating each data bit further comprises the steps of:

assigning each one of the subkeys to a corresponding one of the data bits of the substitute data word; and

mapping each data bit of the substitute data word to a corresponding one of the data bits of the permutated data word using the corresponding assigned subkey.

20. (New) The method of claim 19, where the step of mapping comprises the following steps:

a) selecting a first group of the data bits of the substitute data word as determined by a first one of the plurality of key bits of the corresponding assigned subkey;

b) selecting a second group of the data bits of the substitute data word from the first group of the data bits as determined by a second one of the plurality of key bits of the corresponding assigned subkey; and

c) repeating step b), each time using an additional one of the plurality of key bits of the corresponding assigned subkey until there exists one remaining data bit of the substitute data word, where the one remaining data bit corresponds to the data bit of the substitute data word mapped to the corresponding data bit of the permutated data word.